Pembelajaran Dalam (Deep Learning)

Chapter 4
CNN (Convolutional Neural Networks)

Genap TA 2022/2023

FYS, ITQ | Maret 2023

Convolutional Neural Network (CNN)

1. Pendahuluan Pembelajaran Mesin dan Pembelajaran Dalam (CLO 1)
2. Programming Python: Dasar Numpy, Pandas, Matpotlib, dll (CLO 3)
3. Dasar Regresi dan Klasifikasi (CLO 1)
4. Artificial Neural Networks (ANN) (CLO 2/3)
5. Multi-Layer Perceptron (MLP) (CLO 2/3)
6. TensorFlow (CLO 3)
7. TensorFlow (+Kinerja Machine Learning) (CLO 3/1)
8. TensorFlow Lanjutan (CLO 3)
9. **Convolutional Neural Network (CNN) (CLO 2/3)**
10. CNN Lanjutan (CLO 2/3)
11. Recurrent Neural Networks (RNN) (CLO 2/3)
12. RNN Lanjutan (CLO 2/3)
13. Generative Adversarial Networks (GANs) (CLO 2/3)
14. GANs lanjutan (CLO 2/3)
15. Aplikasi Object Detections (CLO 2/3)
16. UAS (CLO 2/3)

Convolutional Neural Network (CNN)

Course Learning Outcomes (CLO)

- CLO 3: Mahasiswa dapat mengimplementasikan algoritma pembelajaran dalam yang populer menggunakan perangkat lunak

Convolutional Neural Network (CNN)

Tujuan Perkuliahan

- Memahami salah satu metode DNN yaitu CNN
- Mempelajari operasi yang dilakukan di setiap layer CNN.
- Mengimplementasikan CNN dengan librari **TensorFlow.**

Convolutional Neural Network (CNN)

Outline:

A. Memahami CNN dan Mempelajari Hierarki Fitur

- Ektraksi fitur pada DNN
- Hirarki Fitur Di DNN
- Arsitektur CNN

B. Operasi Discreate convolution

- Operasi sebuah Konvolusi diskrit discrete convolution di array satu dimensi.
- Menentukan ukurant dari output Convolution.
- Operasi sebuah discrete convolution di array dua dimensi.

C. Operasi Subsampling (Pooling).

D. Menyatukan semuanya untuk membangun sebuah CNN

- Bekerja dengan beberapa input atau channel warna.
- Convolution Layer untuk multiple channel.
- Convolution Layer untuk multiple feature map

E. Implementasi sebuah deep convolutional neural networks menggunakan TensorFlow

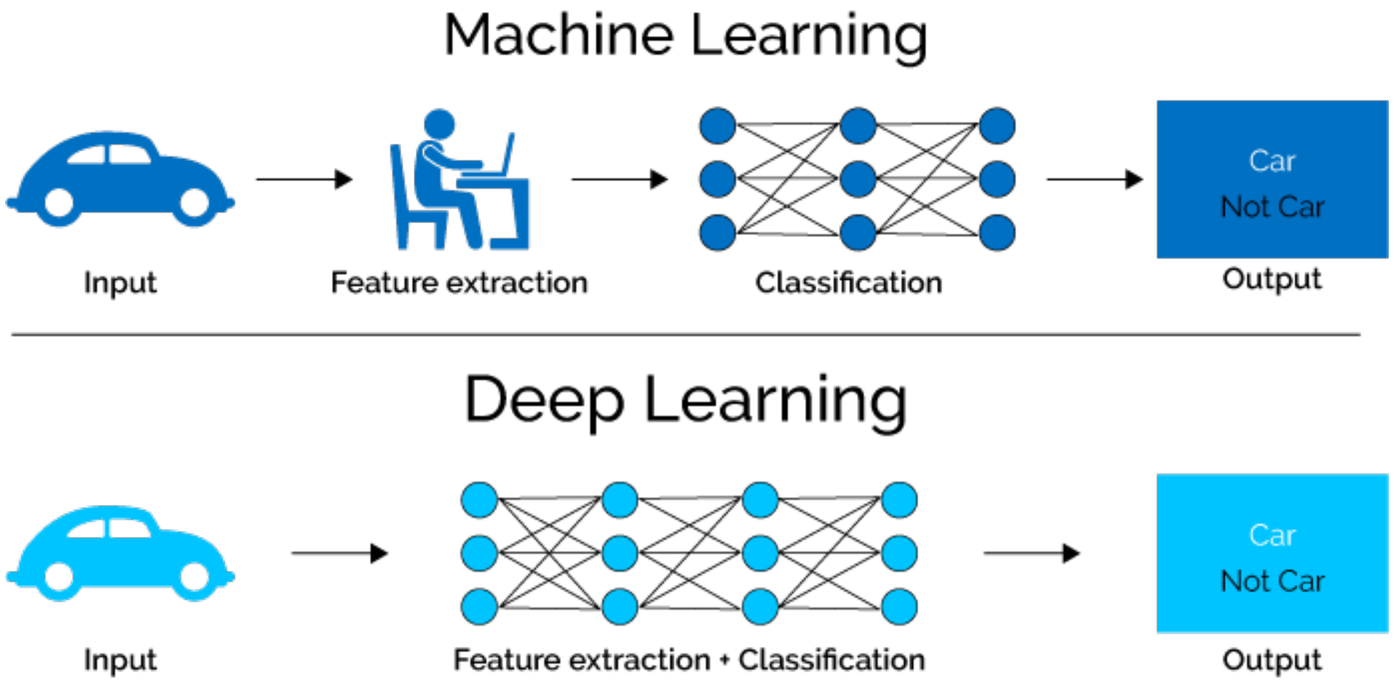
Convolutional Neural Network (CNN)

A. Memahami CNN dan Mempelajari Hierarki Fitur

Ekstraksi fitur poin utama pendukunng Kinerja Machine Learning.

Convolutional Neural Network (CNN)

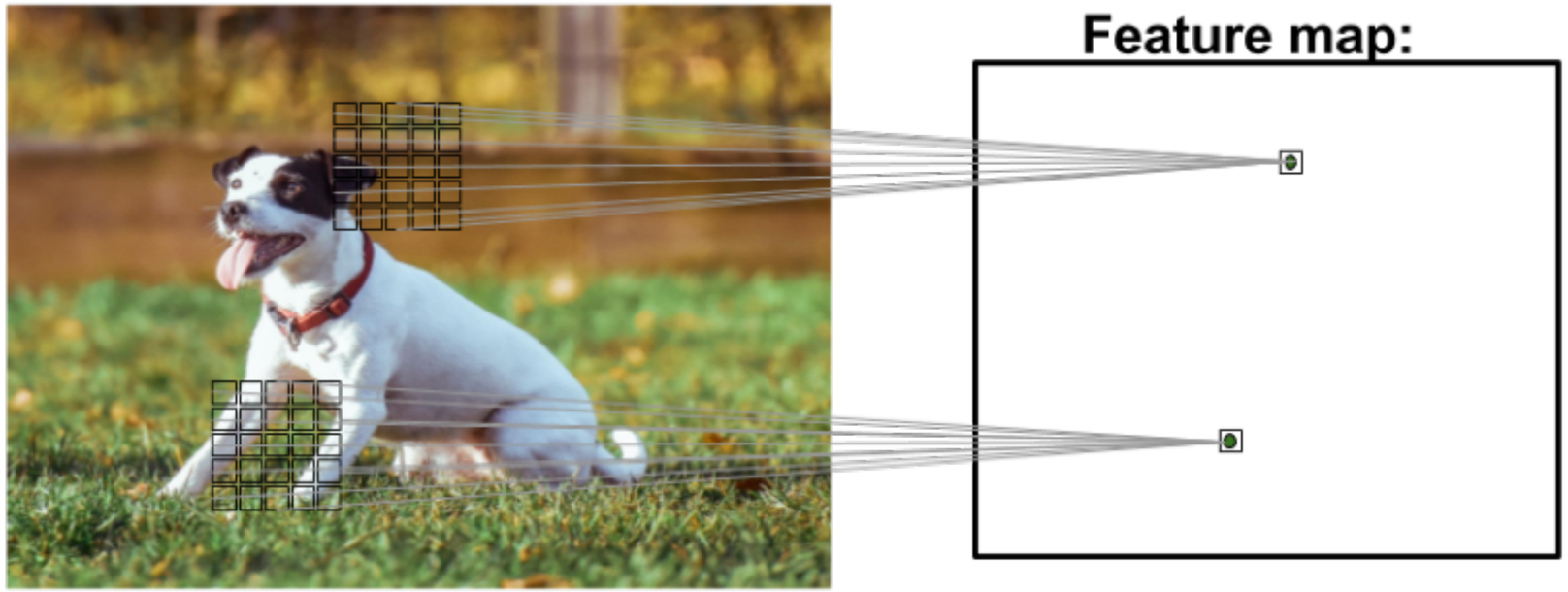
A.1. Ekstraksi Fitur pada DNN



- Berhasil mengekstraksi fitur yang menonjol (relevan) adalah kunci untuk kinerja setiap algoritma machine learning.
- Neural networks memiliki kemampuan dapat secara otomatis mempelajari fitur dari raw data.
- Neural networks dianggap sebagai mesin ekstraksi fitur: lapisan awal mengekstraksi fitur tingkat rendah.

Convolutional Neural Network (CNN)

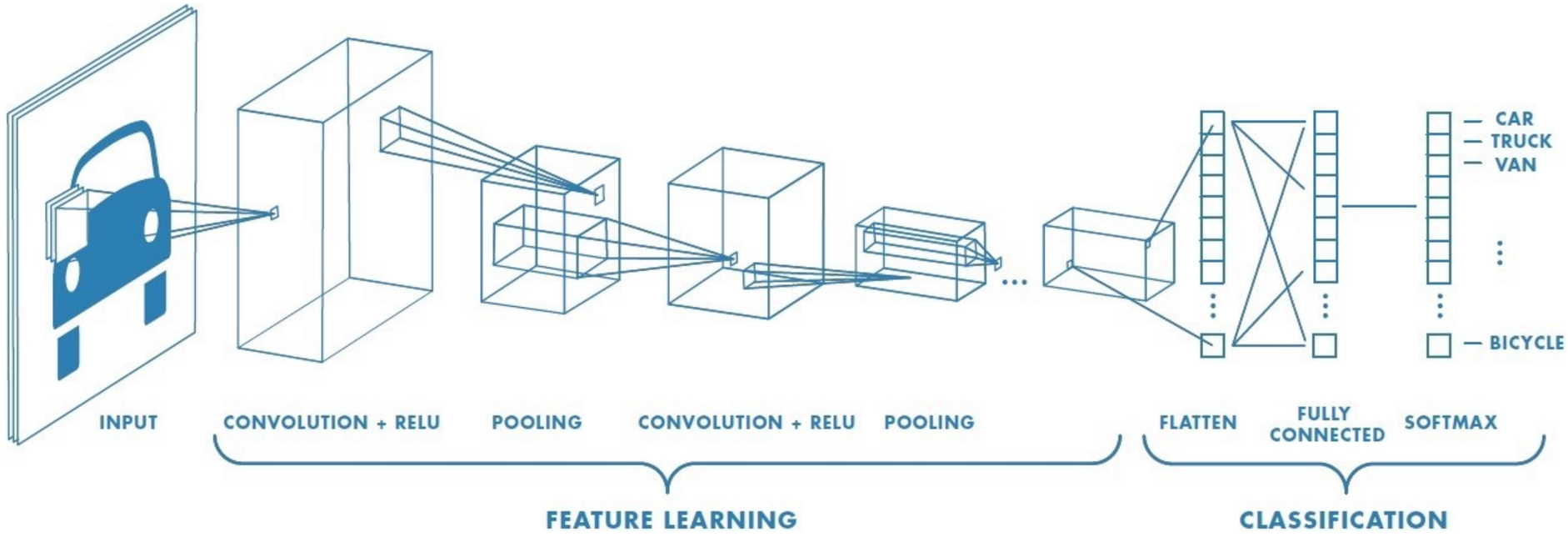
A.2. Hirarki Fitur CNN



- Multilayer neural network, dan khususnya, deep convolutional neural networks, membangun apa yang disebut hierarki fitur dengan menggabungkan fitur tingkat rendah dalam mode lapisan untuk membentuk fitur tingkat tinggi.
- Misalnya, jika kita berurusan dengan gambar, maka fitur tingkat rendah, seperti tepi dan blob, diekstraksi dari lapisan sebelumnya, yang digabungkan bersama untuk membentuk fitur tingkat tinggi, seperti bentuk objek seperti gedung, mobil, atau seekor anjing.

Convolutional Neural Network (CNN)

A.3. Arsitektur CNN



Biasanya, CNN terdiri dari beberapa **Convolutional layer (conv)** dan **subsampling (juga dikenal sebagai Pooling (P))** yang diikuti oleh satu atau lebih **Fully Connected Neural Network(FC) layer di bagian akhir.**

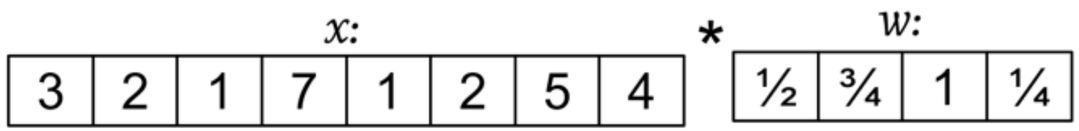
Convolutional Neural Network (CNN)

Bagaimana detail process di setiap komponen arsitektur CNN?

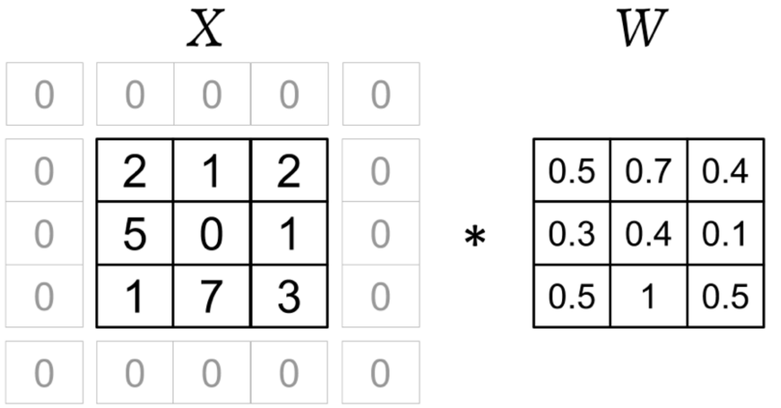
Convolutional Neural Network (CNN)

B. Operasi Discreate convolution

- Sebuah discrete convolution (atau sederhananya convolution) adalah operasi mendasar dalam CNN. Oleh karena itu, penting untuk memahami cara kerja operasi ini.
- Convolution didefinisikan sebagai cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka yang ketiga.
- Mempelajari definisi matematis dan membahas beberapa algoritma dasar untuk menghitung convolution dari dua vektor satu dimensi atau dua matriks dua dimensi.



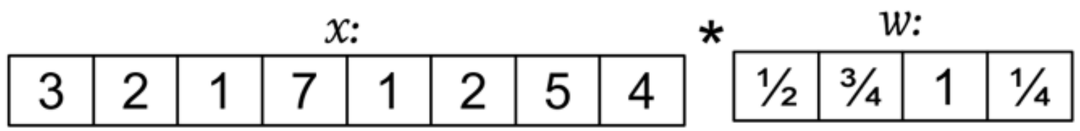
Dua vektor satu dimensi



Dua matriks dua dimensi

Convolutional Neural Network (CNN)

B.1. Operasi sebuah discrete convolution di array satu dimensi



Dua vektor satu dimensi

$$y = x * w \rightarrow y[i] = \sum_{k=-\infty}^{\infty} x[i-k]w[k]$$

- y = x * w , di mana vektor x adalah input dan w disebut filter atau kernel. Konvolusi diskrit secara matematis didefinisikan sebagai rumus diatas.

- Aturan tambahan adalah penambahan padding (nilai 0 di sebelum indek awal atau akhir) untuk input x untuk mengisi nilai lain selain data hingga (finite data) dan stride untuk seberapa jauh operasi pergeseran input untuk operasi convolution untuk setia outputnya.

$X^{padding}(X^p) =$ 0 0 3 2 1 7 1 2 5 4 0 0

Padding = 2

Convolutional Neural Network (CNN)

B.1. Operasi sebuah discrete convolution di array satu dimensi

$$y = x * w \rightarrow y[i] = \sum_{k=-\infty}^{\infty} x[i-k]w[k]$$

- Ada dua hal aneh dalam rumus diatas yang perlu kita perjelas: -∞ ke ∞ indeks dan pengindeksan negatif untuk x.

- Masalah pertama di mana penjumlahan dilakukan melalui indeks dari -∞ ke ∞ tampak aneh terutama karena dalam aplikasi pembelajaran mesin, yang selalu berurusan dengan vektor fitur hingga/ *finite.* Sehingga operasi convolution dibatasi sesuai jumlah fitur input. Sehingga persamaan menjadi seperti berikut:

$$y = x * w \rightarrow y[i] = \sum_{k=-0}^{k=m-1} x^p[i+m-k]w[k]$$

Convolutional Neural Network (CNN)

B.1. Operasi sebuah discrete convolution di array satu dimensi

$$y = x * w \rightarrow y[i] = \sum_{k=-\infty}^{\infty} x[i-k]w[k]$$

- Ada dua hal aneh dalam rumus diatas yang perlu kita perjelas: -∞ ke ∞ indeks dan pengindeksan negatif untuk x.

- Masalah pertama di mana penjumlahan dilakukan melalui indeks dari -∞ ke ∞ tampak aneh terutama karena dalam aplikasi pembelajaran mesin, yang selalu berurusan dengan vektor fitur hingga/ *finite.* Sehingga operasi convolution dibatasi sesuai jumlah fitur input. Sehingga persamaan menjadi seperti berikut. m dalah dimensi input, dimana operasi akan berlangsung dari k = 0 sampan K = m-1.

$$y = x * w \rightarrow y[i] = \sum_{k=-0}^{k=m-1} x^p[i+m-k]w[k]$$

Convolutional Neural Network (CNN)

B.1. Operasi sebuah discrete convolution di array satu dimensi

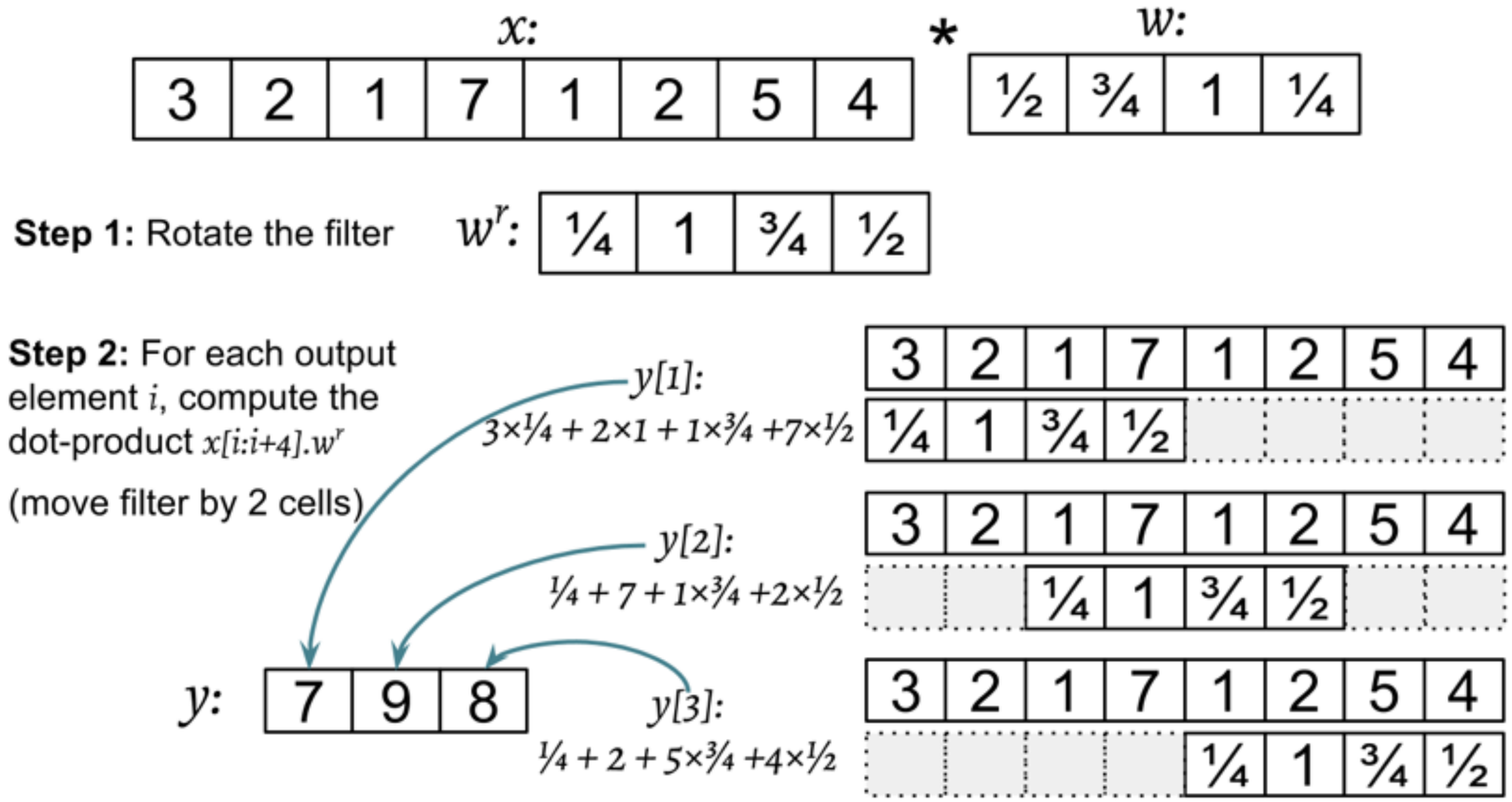
$$y = x * w \rightarrow y[i] = \sum_{k=-0}^{k=m-1} x^p[i+m-k]w[k]$$

- Masalah kedua adalah bahwa x dan w diindeks dalam arah yang berbeda dalam penjumlahan ini.
- Untuk masalar ini, kita dapat membalikkan salah satu vektor tersebut, x atau w, setelah diberikan padding. Kemudian, kita cukup menghitung perkalian titik mereka.
- Mari kita asumsikan kita membalik filter w untuk mendapatkan filter w^r yang diputar. Kemudian, dot product dihitung $X[i:i+m] \,.\, w^r$ untuk mendapatkan satu elemen $y[i]$, dimana $x[i:i+m]$ adalah blok input x dengan ukuran m.
- Operasi ini diulangi seperti pada pendekatan jendela geser untuk mendapatkan semua elemen output.

Convolutional Neural Network (CNN)

B.1. Operasi sebuah discrete convolution di array satu dimensi

Gambar berikut memberikan contoh dengan $x = (3,2,1,7,1,2,5,4)$ dan $w = \left(\frac{1}{2}, \frac{3}{4}, 1, \frac{1}{4}\right)$ sehingga tiga elemen output pertama dihitung.



Convolutional Neural Network (CNN)

B.2. Menentukan ukurant dari output Convolution

- Ukuran output dari sebuah convolution ditentukan oleh jumlah total kali process menggeser filter *w* sepanjang vektor input. Misalkan vektor input berukuran *n* dan filter berukuran *m*.
- Kemudian, ukuran output yang dihasilkan dari peynambahan padding *p* dan pergeseran window dot product / stride *s* ditentukan sebagai berikut:

$$o = \left[\frac{n + 2p - m}{s}\right] + 1$$

Contoh:Hitung ukuran output untuk vektor input ukuran (*n*)10 dengan kernel convolution (w) ukuran (*w*) 5, padding (*p*) 2, dan stride (*s*) 1:

$$n = 10, m = 5, p = 2, s = 1 \rightarrow o = \left[\frac{10 + 2 \times 2 - 5}{1}\right] + 1 = 10$$

Convolutional Neural Network (CNN)

B.3. Operasi sebuah discrete convolution di array dua dimensi

- Konsep yang dipelajari di bagian sebelumnya diperluas ke dua dimensi. Ketika kita berurusan dengan input dua dimensi, seperti matriks $X_{n_1 \times n_2}$ dan matriks filter $W_{m_1 \times m_2}$, di mana $m_1 \leq n_1$ dan $m_2 \leq n_2$, maka matriks $Y = X * W$ tersebut adalah hasil convolution 2D dari X dengan W. Ini secara matematis didefinisikan sebagai berikut:

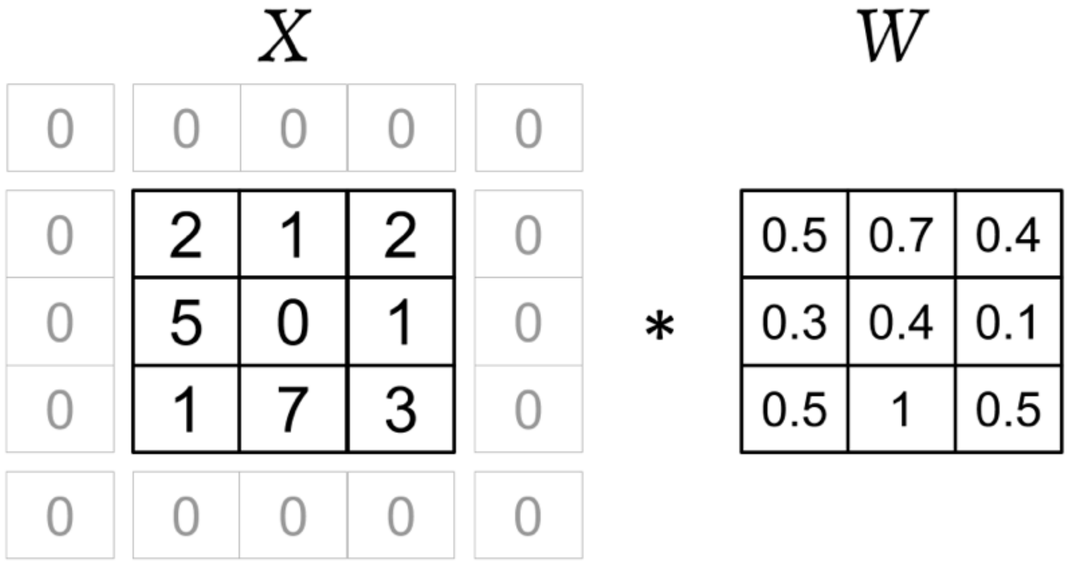
$$Y = X * W \rightarrow y[i,j] = \sum_{k_1=-\infty}^{+\infty} \sum_{k_2=-\infty}^{+\infty} x[i-k_1, j-k_2] w[k_1, k_2]$$

- Perhatikan bahwa jika kita menghilangkan salah satu dimensi, rumus yang tersisa sama persis dengan rumus yang digunakan sebelumnya untuk menghitung convolution dalam 1D.
- Faktanya, semua teknik yang disebutkan sebelumnya, seperti zero-padding, memutar matriks filter, dan penggunaan strides, juga berlaku untuk convolution 2D.

Convolutional Neural Network (CNN)

B.3. Operasi sebuah discrete convolution di array dua dimensi

- Contoh berikut mengilustrasikan perhitungan convolution 2D antara matriks input $X_{3\times3}$, matriks kernel $W_{3\times3}$, padding $P = (1,1)$, dan stride $S = (2,2)$ Menurut padding yang ditentukan, satu lapisan nol diisi pada setiap sisi matriks input, yang menghasilkan matriks padding $X_{3\times3}^{padded}$, sebagai berikut:



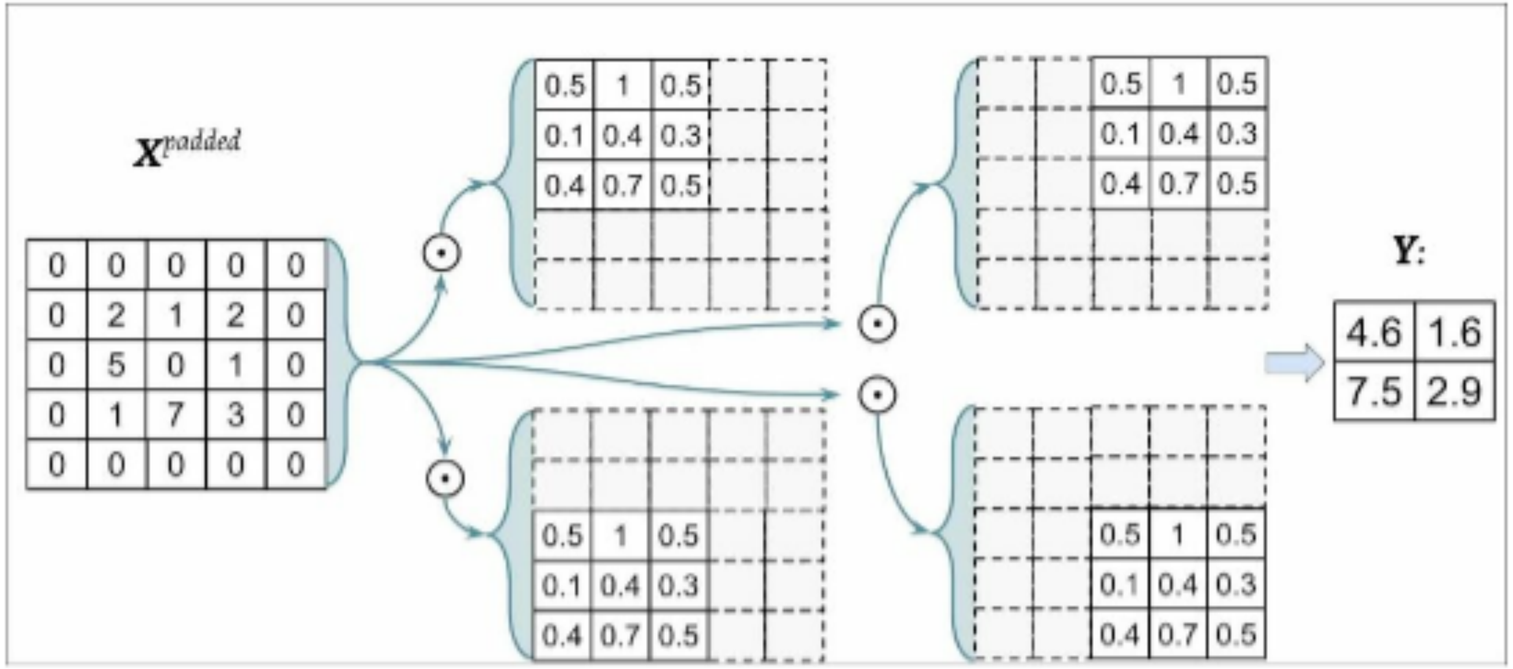
Convolutional Neural Network (CNN)

B.3. Operasi sebuah discrete convolution di array dua dimensi

- Tahap awal rotasi terlebih dahulu putar/rotasi matrik filter, seperti berikat:

$$W^r = \begin{bmatrix} 0.5 & 1 & 0.5 \\ 0.1 & 0.4 & 0.3 \\ 0.4 & 0.7 & 0.5 \end{bmatrix}$$

  - Kemudiam melakukan dot product input dan matrik filter seperti berikat:



Convolutional Neural Network (CNN)

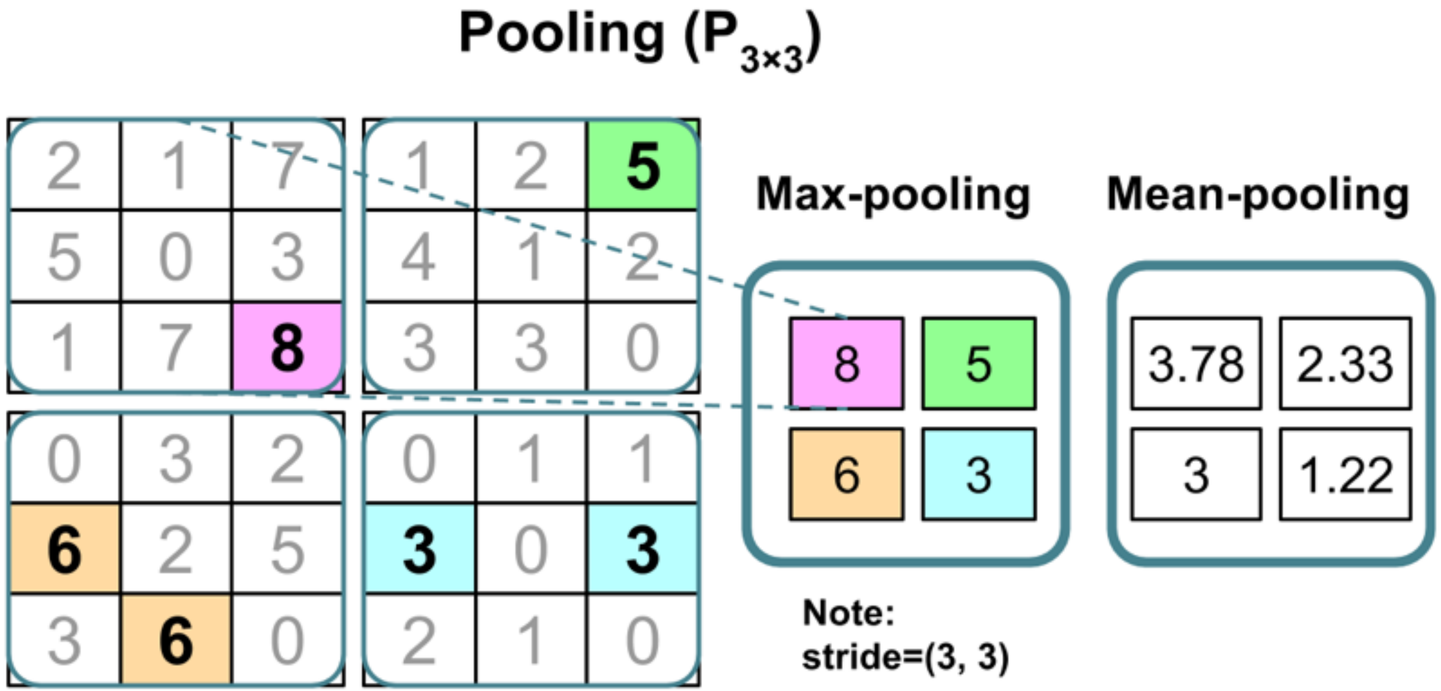
C. Operasi Subsampling (Pooling)

- subsampling biasanya diterapkan dalam dua bentuk operasi pooling di convolutional neural network : **max-pooling dan mean-poolin (juga dikenal sebagai average-pooling)**.
- Pooling layer biasanya dilambangkan dengan $P_{n_1 \times n_2}$. Di sini, subskrip menentukan ukuran neighborhood (jumlah piksel yang berdekatan di setiap dimensi), dimana operasi maks atau rata-rata dilakukan. Kami menyebut neighborhood seperti itu sebagai ukuran pooling.

Convolutional Neural Network (CNN)

C. Operasi Subsampling (Pooling)

- Gambar dibawah ini mendeskripsikan operasi max pooling. Di sini, max-pooling mengambil nilai maksimum dari neighborhood piksel, dan mean-pooling menghitung ra



Convolutional Neural Network (CNN)

D. Menyatukan semuanya untuk membangun sebuah CNN

- Dalam convolutional neural networks, operasi$a = Wx + b$ pada MLP digantikan oleh operasi convolutional, seperti di mana X adalah matriks yang mewakili piksel dalam susunan tinggi x lebar.
- Dalam kedua kasus, pra-aktivasi diteruskan ke fungsi aktivasi untuk mendapatkan aktivasi dari sebuah unit hidden layer $H = \phi(A)$ disetiap convolution layer, dimana ϕ adalah fungsi aktivasi layaknya MLP.
- Selain itu, ingatlah bahwa subsampling adalah blok bangunan lain dari convolutional neural networks, yang mungkin muncul dalam bentuk pooling, seperti yang telah kami jelaskan di bagian sebelumnya.

Convolutional Neural Network (CNN)

D.1. Bekerja dengan beberapa input atau channel warna

- Sampel input ke Convotional layer dapat berisi satu atau lebih dari 2D matriks array dengan dimensi $N_1 \times N_2$ (misalnya, tinggi dan lebar gambar dalam piksel). Dimensi $N_1 \times N_2$ yang merupakan kumpulan matrik disebut **channels**.
- Oleh karena itu, menggunakan **multiple channel** sebagai input ke convotional layer mengharuskan menggunakan tensor rank-3 atau 3D array: $X_{N_1 \times N_2 \times C_{in}}$, C_{in} merupakan jumlah input channel.
- Sebagai contoh, mari kita lihat gambar sebagai input ke lapisan pertama CNN. Jika gambar diwarnai dan menggunakan mode warna RGB, maka $C_{in} = 3$ (untuk saluran warna merah, hijau, dan biru dalam RGB).
- Namun, jika gambar dalam skala abu-abu (grayscale), maka kita memiliki $C_{in} = 1$ karena hanya ada satu channel dengan nilai intensitas piksel skala abu-abu.

Convolutional Neural Network (CNN)

D.2. Convolution Layer untuk multiple channel

- Operasi convolution untuk setiap channel dilakukan secara terpisah dan kemudian menjumlahkan hasilnya menggunakan penjumlahan matriks.
- Convolution yang terkait dengan setiap channel (c) memiliki matriks kernelnya sendiri $W[:,:,c]$. Total hasil pra-aktivasi dihitung dalam rumus berikut:

Given a sample $\mathbf{X}_{n_1 \times n_2 \times c_{in}}$,
a kernel matrix $\mathbf{W}_{m_1 \times m_2 \times c_{in}}$,
and bias value b

$$\Rightarrow \begin{cases} \boldsymbol{Y}^{Conv} = \sum_{c=1}^{C_{in}} \boldsymbol{W}[:,:,c] * \boldsymbol{X}[:,:,c] \\ \text{pre-activation}: \quad \mathbf{A} = \mathbf{Y}^{Conv} + \mathrm{b} \\ \text{Feature map}: \quad \boldsymbol{H} = \phi(\boldsymbol{A}) \end{cases}$$

- Hasil akhir, H, disebut feature map. Biasanya, convolutional layer di CNN memiliki lebih dari satu feature map.

Convolutional Neural Network (CNN)

D.3. Convolution Layer untuk multiple feature map

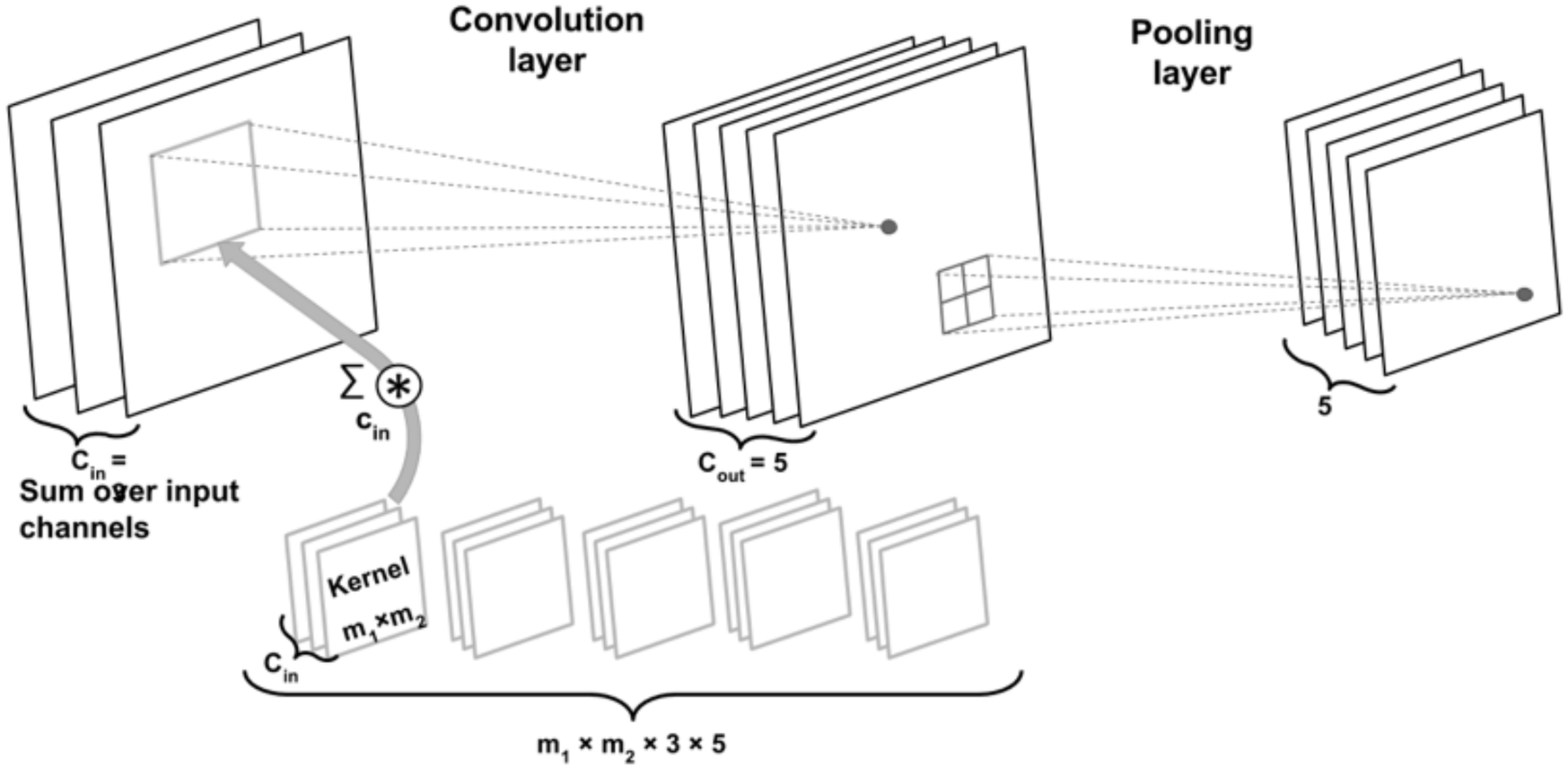
- Jika kita menggunakan multiple feature map, tensor kernel menjadi empat dimensi: $(width \times height \times C_{in} \times C_{out})$. Di sini, $(width \times height)$ lebar x tinggi adalah ukuran kernel, C_{in} jumlah channel input, dan C_{out} jumlah feature map output. Jadi, sekarang mari sertakan jumlah feature map keluaran dalam rumus sebelumnya dan perbarui sebagai berikut:

$$\text{Given a sample } \boldsymbol{X}_{n_1 \times n_2 \times C_{in}} \text{, kernel matrix } \boldsymbol{W}_{m_1 \times m_2 \times C_{in} \times C_{out}} \text{ and bias vector } \boldsymbol{b}_{C_{out}} \Rightarrow \begin{cases} \boldsymbol{Y}^{Conv}[:,:,k] = \sum_{c=1}^{C_{in}} \boldsymbol{W}[:,:,c,k] * \boldsymbol{X}[:,:,c] \\ \boldsymbol{A}[:,:,k] = \boldsymbol{Y}^{Conv}[:,:,k] + \boldsymbol{b}[k] \\ \boldsymbol{H}[:,:,k] = \phi(\boldsymbol{A}[:,:,k]) \end{cases}$$

Convolutional Neural Network (CNN)

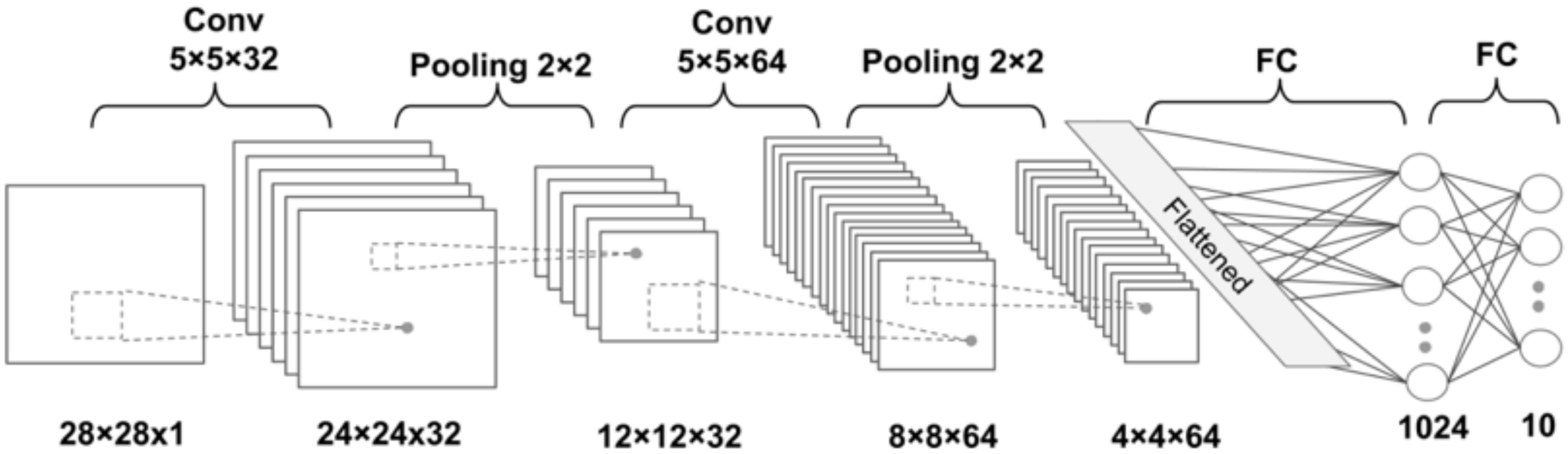
D. Menyatukan semuanya untuk membangun sebuah CNN

- Dalam contoh ini, ada tiga channel input. Tensor kernel adalah empat dimensi. Setiap matriks kernel dilambangkan sebagai $m_1 \times m_2$, dan ada tiga array 2D di antaranya, satu untuk setiap channel input. Selain itu, ada lima kernel seperti itu, terhitung untuk lima feature map keluaran. Terakhir, ada pooling layer untuk membuat subsampling peta fitur, seperti yang ditunjukkan pada gambar berikut:



Convolutional Neural Network (CNN)

E. Implementasi sebuah deep convolutional neural networks menggunakan TensorFlow

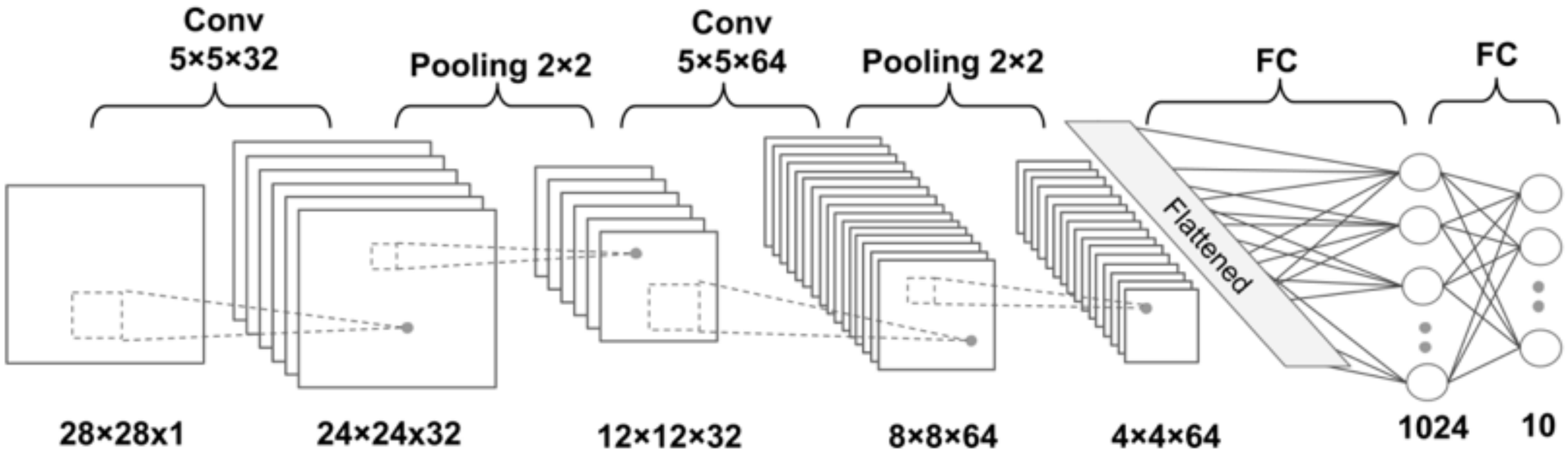


Dimensi tensor di setiap lapisan adalah sebagai berikut:

1. Input: $[batchsize \times 28 \times 28 \times 1]$
2. Conv_1: $[batchsize \times 24 \times 24 \times 32]$
3. Pooling_1: $[batchsize \times 12 \times 12 \times 32]$
4. Conv_2: $[batchsize \times 8 \times 8 \times 64]$
5. Pooling_1: $[batchsize \times 4 \times 4 \times 64]$
6. FC_1: $[batchsize \times 1024]$
7. FC_2 dan softmax layer: $[batchsize \times 10]$

Convolutional Neural Network (CNN)

E. Implementasi sebuah deep convolutional neural networks menggunakan TensorFlow



Implementasi code dijelaskan di hands on CNN dan Latihan code yang diberikan.

Thank You